Local Stability and Convergence Analysis of Neural Network Controllers with Error Integral Inputs

Xingang Fu, *Member, IEEE,* Shuhui Li, *Senior Member, IEEE,* Donald C. Wunsch, *Fellow, IEEE,* and Eduardo Alonso

***Abstract*—This paper investigates the local stability and convergence for a class of neural network controllers with error integral as inputs for reference tracking. It is proved that if the neural network controllers only have error terms as inputs, the control system has a non-zero steady-state error for any constant reference except for one special point. It is proved that adding error integral to the inputs of the neural network controller is one sufficient way to remove steady-state error for any constant reference. Due to the nonlinearity of the neural network controllers, the neural network control systems are linearized at the equilibrium points. It is proved that if all the eigenvalues of the linearized neural network control systems have negative real parts, local asymptotic stability and local exponential convergence are guaranteed. Two case studies were explored to verify the theoretical proofs: a single-layer neural network controller in a one-dimensional system and a four-layer neural network controller in a two-dimensional system applied in renewable energy integration. Simulations have demonstrated that when neural network controllers and corresponding Generalized Proportional-Integral controllers have the same eigenvalues, all control systems exhibit almost the same transient responses in a small enough neighborhood of their respective equilibrium points.**

***Index Terms*—Neural Network Controller, Error Integral, Steady-State Error, Local Asymptotic Stability, Local Exponential Convergence, Generalized PI controller.**

I. INTRODUCTION

RECENTLY, Dynamic Programming (DP) [1] has been used extensively for the optimal control of nonlinear systems [2], [3], [4]. As one type of Approximate Dynamic Programming (ADP) method, Adaptive Critic Designs (ACD) have been adopted to approximate the optimal cost and optimal control of a system [5], [6], [7], [2]. In [8], [9], a neural network (NN) was trained based on the ADP principle to control a three-phase Inductor (L) filter-based Grid-Connected Converter (GCC) system. The ADP-based NN controlling of the Inductor-Capacitor-Inductor (LCL) filter-based three-phase [10] and single-phase [11] GCC systems was also demonstrated to be able to yield an excellent performance compared to the conventional Proportional-Integral (PI) controller based control methods. In [10], it has been demonstrated that Recurrent Neural Network (RNN) vector control has a wider stability region for the system parameter change than Active Damping (AD) or Passive Damping (PD) vector control for LCL-based grid-connected converter systems. [12] implemented a NN vector controller with error integral inputs in a permanent-magnet synchronous motor (PMSM) to overcome the decoupling inaccuracy problem associated with the conventional PI-based vector-control methods.

Even though the NN controller has many successful applications, the theoretical foundation or analysis is missing. Normally, the NN is considered as the black-box technique[13], not the white-box models. When the NN is applied to a real application, however, many issues have arisen, e.g. the stability problem, the convergence problem. The stability problem is very critical in the neural control system[14], [15] especially when applied in the real system. When the system is unstable, the output of the system is out of control, which could cause serious damage to the system. This is the problem that curbs the applications of NN controllers into real life by control engineers and electric engineers. How to guarantee the training of an NN will converge is a difficult problem[16]. Towards which direction, the training will be more effective and faster is still unsolved.

This research work specifically intends to study the local stability and local convergence of NN controllers with error integral terms. The specific contributions of the paper are listed as follows: 1) a proof that adding error integral to the inputs of the NN controller is sufficient to remove the steady-state error of the NN controller for any constant reference, 2) obtaining the condition requirement of NN controllers to guarantee the local asymptotic stability and local exponential convergence, which is that all eigenvalues of the neural network control systems should have negative real parts, 3) revealing that the NN controllers and the corresponding generalized PI controllers having the same eigenvalues should have exactly the same responses in a small enough domain of their respective equilibrium points in the steady state, and 4) case studies of a one-dimensional NN controller and a two-dimensional four-layer NN controller applied in renewable energy integration for verifying theoretical theorems.

The rest of the paper is structured as follows. Section II studies the local stability and local convergence of the single-layer NN controller of two structures: one structure with only error terms and the other one with error terms and error integral terms. The local stability and local convergence analysis of general form multi-layer NN controllers of two structures: one structure with only error terms and the other one with error terms and error integral terms are investigated in

Xingang Fu is with the Department of Electrical Engineering and Computer Science, Texas A&M Kingsville, Kingsville, TX 78363, USA (email: Xingang.Fu@tamuk.edu).

Shuhui Li is with the Department of Electrical and Computer Engineering, The University of Alabama, Tuscaloosa, AL 35487, USA (email: sli@eng.ua.edu).

Donald C. Wunsch, is with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA (email: dwunsch@mst.edu).

Eduardo Alonso is with the Artificial Intelligence Research Centre (CitAI), City, University of London, EC1V 0HB, UK (email: e.alonso@city.ac.uk).

Section III. A case study for a one-dimensional NN controller is demonstrated in Section IV to verify the conclusions of Section II. Section V investigates a four-layer NN controller in a two-dimensional system applied in electric power applications to verify the theoretical conclusion of Section III. Finally, the paper concludes with summary remarks in section VII.

II. Single-Layer Neural Network Controllers

Consider the following time-invariant state-space model

$$\dot{x} = Ax + Bu \tag{1}$$

where, x is the system state vector, $\dot{x}$ denotes the derivative of the state vector x with respect to time t, u stands for input or control vector, and A and B are system matrix and input matrix respectively, with $A \neq 0$ and $B \neq 0$.

A. NN Controllers with Only Error Term Inputs

If a single-layer NN controller has only the error term e as input, the control vector u can be expressed as

$$u = k\tanh(we + c) \tag{2}$$

in which, w and c represent the weight matrix and the bias vector of the NN controller, the constant scalar k stands for an actuator gain, and the error e is defined as

$$e = x_{ref} - x \tag{3}$$

with x_{ref} representing the reference for the system state x.

According to the definition of the error term e in (3), the following two equations hold true.

$$x = x_{ref} - e \tag{4}$$
$$\dot{e} = -\dot{x} \tag{5}$$

We can now substitute (2), (4), and (5) into (1), and rewrite (1) into the closed-loop system with tracking error e as the system state, as follows:

$$\dot{e} = f(e) = A(e - x_{ref}) - kB\tanh(we + c) \tag{6}$$

Theorem 1. *For a neural dynamic system (6), $e = 0$ is not an equilibrium point except when $x_{ref} = x^*_{ref} = -\frac{1}{k}A^{-1}B\tanh(c)$. The system will have a non-zero steady-state error, that is $e(\infty) \neq 0$, for any constant reference except x^*_{ref}.*

Proof: The equilibrium point of (6) is the root of the function $f(e)$. If we substitute $e = 0$ into (6), the function $f(e)$ equals

$$f(e) = -Ax_{ref} - kB\tanh(c) \neq 0. \tag{7}$$

Only when $x_{ref} = x^*_{ref} = -\frac{1}{k}A^{-1}B\tanh(c)$, $f(e) = 0$. Thus, $e = 0$ is not an equilibrium point of system (6) except for one special reference point x^*_{ref}.

We denote e^* to represent the root of $f(e)$, which satisfies the following equation

$$f(e^*) = A(e^* - x_{ref}) - kB\tanh(we^* + c) = 0 \tag{8}$$

Thus e^* is the equilibrium point of (6) and $e^* \neq 0$, which also means the system has a non-zero steady-state error $e(\infty) = e^*$. ∎

Lemma 1. *For the linear time-invariant system*

$$\dot{x} = Gx \tag{9}$$

with a constant system matrix G, if all eigenvalues of G have the negative real parts, the equilibrium point $x = 0$ is globally asymptotic stable and the exponential convergence is also guaranteed.

Proof: As (9) is a set of first-order homogeneous linear differential equations with constant coefficients and system matrix G is constant, G has n linearly independent eigenvectors, the analytical solution has the following form

$$e_n(t) = \sum_{i=1,\ldots,n} c_i exp^{\lambda_i t}\nu_i \tag{10}$$

where, the vector λ_i are eigenvalues of G, ν_i are eigenvectors of G, c_i are some constants determined by the initial condition of the system and exp represents the base of the natural logarithm.

The eigenvalues of G can be determined by the following equation

$$\det(\lambda I - G) = 0 \tag{11}$$

where λ denotes the eigenvalue and I is the identity matrix.

When all eigenvalues have the negative real parts, that is $Re(\lambda_i) < 0$, $\lim_{t\to\infty} e_n = 0$, which is also exponentially convergent ([17], [18], [19], [20]). ∎

Theorem 2. *For a neural dynamic system (6), local asymptotic stability and local exponential convergence are guaranteed if the weight matrix w and bias vector c of the NN controller satisfy the following condition*

$$Re\left\{\mathrm{eig}\left(A - kBw\,\mathrm{diag}(1 - \tanh^2(we^* + c))\right)\right\} < 0 \tag{12}$$

in which, eig *denotes the eigenvalue operator, Re stands for the real part,* diag *represents diagonal matrix operator, and e^* is the equilibrium point of (6).*

Proof: The equilibrium point of (6) can be shifted from e^* to 0 by defining a new variable e_n

$$e_n = e - e^* \tag{13}$$

and thus

$$\dot{e_n} = \dot{e} \tag{14}$$

If we substitute (13) and (14) into (6), the new system equation will be

$$\dot{e_n} = f(e_n) = A(e_n + e^* - x_{ref}) - kB\tanh(w(e_n + e^*) + c) \tag{15}$$

For (15), the equilibrium point of the system is $e_n = 0$.

The right side of (15) are nonlinear functions. Under the definition of Lyapunov stability [18], we can use the first-order derivative to linearize the system at $e_n = 0$ and obtain the following set of linear equations

$$\dot{e_n} = (\frac{\partial f}{\partial e_n}|_{e_n=0})e_n = Ge_n \tag{16}$$

where the system matrix G equals

$$G = A - kBw\,\mathrm{diag}(1 - \tanh^2(we^* + c)) \quad (17)$$

According to Lemma 1, as long as all the eigenvalues of G have the negative real parts, that is $Re\{\mathrm{eig}(G)\} < 0$, the system asymptotic stability and exponential convergence are guaranteed. However, as the system (16) is linearized at the equilibrium point, only the local asymptotic stability and local exponential convergence can be guaranteed. ∎

Remark 1. In (12), the reference x_{ref} does not exist explicitly. However, e^* are the roots of (6). When all system parameters (A, B, k) and the NN weight w and bias c are kept unchanged, e^* depends on x_{ref}. So, the system matrix G and eigenvalues λ_i are implicit functions of x_{ref} and thus the reference x_{ref} affects the stability of the system.

Corollary 2.1. *Consider a generalized proportional controller $u = kK_pe$ with the constant proportional gain matrix K_p, which can be regarded as a special case of the single-layer NN controller with a linear identity function as the activation function. Thus the steady-state error $e(\infty)$ and the equilibrium point e^* are*

$$e(\infty) = e^* = (A - kBK_p)^{-1}Ax_{ref} \quad (18)$$

and the corresponding global stability condition is

$$Re\{\mathrm{eig}(A - kBK_p)\} < 0 \quad (19)$$

The reference x_{ref} is not contained in (19) and thus will not affect the system stability.

B. NN Controllers with Error Integral Inputs

Consider a single-layer NN controller having error e and error integral s as inputs. The control vector u is expressed as

$$u = k\tanh(we + vs + c) \quad (20)$$

in which, the error integral s is defined as

$$s = \int_0^t e(\tau)d\tau \quad (21)$$

If we substitute (4), (5), and (20) into (1), the system equation will be simplified as

$$\dot{e} = A(e - x_{ref}) - kB\tanh(we + vs + c) \quad (22)$$

From the definition of error integral s in (21), the following equation can be derived

$$\dot{s} = e \quad (23)$$

Thus combining (22) and (23), a new augmented state-space model can be obtained

$$\begin{cases} \dot{e} = f_1(e,s) = A(e - x_{ref}) - kB\tanh(we + vs + c) \\ \dot{s} = f_2(e,s) = e \end{cases} \quad (24)$$

Through this conversion, the original n-dimension NN control system (22) is converted into a $2n$-dimensional system (24).

Remark 2. This conversion is not an equivalent transformation. From (21), (23) can be derived. However, from (23), (21) is not the only solution. In general, many solutions can be obtained from (23) and the general solution is

$$s = \int_0^t e(\tau)d\tau + C \quad (25)$$

in which, C is one constant vector.

Theorem 3. *For a neural dynamic system (24), $e = 0$ is an equilibrium point and the system does not have a steady-state error for any constant reference x_{ref}.*

Proof: The equilibrium point of (24) are the roots of the right side function, that is

$$\begin{cases} f_1(e,s) = A(e - x_{ref}) - kB\tanh(we + vs + c) = 0 \\ f_2(e,s) = e = 0 \end{cases} \quad (26)$$

From the second equation of (26), e must be 0. Thus the equilibrium point will be (0, s^*), in which s^* equals

$$s^* = -v^{-1}[\mathrm{arctanh}(\frac{1}{k}B^{-1}Ax_{ref}) + c] \quad (27)$$

The equilibrium point of (24) is (0, s^*), which means the system error e converges to 0 while the error integral s converges to s^* when the time goes to infinity. When there is an error integral term s feeding into the input of the NN controller, it is guaranteed that there is no steady-state error in the system. ∎

Theorem 4. *For a neural dynamic system (24), local asymptotic stability and local exponential convergence are guaranteed if the weight matrix w and v of the NN controller satisfy the following condition*

$$Re\left\{\mathrm{eig}\left(\begin{bmatrix} G_{11} & G_{12} \\ I & 0 \end{bmatrix}\right)\right\} < 0 \quad (28)$$

where G_{11} and G_{12} equal

$$G_{11} = A - kBw\,\mathrm{diag}(1 - (\frac{1}{k}B^{-1}Ax_{ref})^2) \quad (29)$$

$$G_{12} = -kBv\,\mathrm{diag}(1 - (\frac{1}{k}B^{-1}Ax_{ref})^2) \quad (30)$$

Proof: The equilibrium point of (24) can be shifted from [0; s^*] to [0; 0] using the following conversion:

$$\begin{cases} e = e \\ s_n = s - s^* \end{cases} \quad (31)$$

Substituting (31) into (24), the new augmented system equation will be

$$\begin{cases} \dot{e} = f_1(e,s_n) = A(e - x_{ref}) - kB\tanh[we + v(s_n + s^*) + c] \\ \dot{s_n} = f_2(e,s_n) = e \end{cases} \quad (32)$$

Under the definition of Lyapunov stability [18] and linearizing (32) at the equilibrium point [0, 0], the system equation will become

$$\begin{bmatrix} \dot{e} \\ \dot{s_n} \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}\begin{bmatrix} e \\ s_n \end{bmatrix} \quad (33)$$

in which, G_{11}, G_{12}, G_{21}, and G_{22} are defined as

$$G_{11} = \frac{\partial f_1(e, s_n)}{\partial e}|_{e=0,s_n=0} = A - kBw\,\mathrm{diag}(1 - \tanh^2(vs^* + c)) \tag{34}$$

$$G_{12} = \frac{\partial f_1(e, s_n)}{\partial s_n}|_{e=0,s_n=0} = -kBv\,\mathrm{diag}(1 - \tanh^2(vs^* + c)) \tag{35}$$

$$G_{21} = \frac{\partial f_2(e, s_n)}{\partial e}|_{e=0,s_n=0} = I \tag{36}$$

$$G_{22} = \frac{\partial f_2(e, s_n)}{\partial s_n}|_{e=0,s_n=0} = 0 \tag{37}$$

From (27), $\tanh(vs^* + c) = -\frac{1}{k}B^{-1}Ax_{ref}$, thus G_{11} and G_{12} can be further simplified as

$$G_{11} = A - kBw\,\mathrm{diag}(1 - (\frac{1}{k}B^{-1}Ax_{ref})^2) \tag{38}$$

$$G_{12} = -kBv\,\mathrm{diag}(1 - (\frac{1}{k}B^{-1}Ax_{ref})^2) \tag{39}$$

As G_{11}, G_{12}, G_{21}, and G_{22} are all constants, according to Lemma 1, if the NN weights w and v satisfy the following condition,

$$Re\left\{\mathrm{eig}\left(\begin{bmatrix} G_{11} & G_{12} \\ I & 0 \end{bmatrix}\right)\right\} < 0 \tag{40}$$

the system asymptotic stability and exponential convergence are guaranteed. However, as the system (33) is linearized at the equilibrium point, only the local asymptotic stability and local exponential convergence can be guaranteed. ∎

Remark 3. Although the bias vector c of NNs is not contained in (29) and (30), c affects the location of s^* from (27) and thus affects the convergence region of the equilibrium point.

Corollary 4.1. *Consider a generalized PI controller $u = k(K_pe + K_is)$, where K_p and K_i are the constant matrices representing the proportional gains and integral gains respectively. This generalized PI controller can be regarded as a special case of the single-layer NN controller with a linear identity function as the activation function. Thus the equilibrium point of the system is (0, s^*) and s^* equals*

$$s^* = -K_i^{-1}(\frac{1}{k}B^{-1}Ax_{ref}) \tag{41}$$

To guarantee global stability and exponential convergence, the following condition needs to be satisfied

$$Re\left\{eig\left(\begin{bmatrix} A - kBK_p & -kBK_i \\ I & 0 \end{bmatrix}\right)\right\} < 0 \tag{42}$$

The reference x_{ref} will not affect the stability and convergence of the control system.

Remark 4. For a single-layer NN controller with only error terms (2) or with error terms and error integral terms (19), the reference x_{ref} will appear in the condition equations (9) and (27) explicitly or inexplicitly, and thus will affect the system stability. So the weights and bias vector of the NN controller together with the reference will determine the system local stability and local convergence.

III. Multi-Layer Neural Network Controllers

In this section, the multi-layer NN controller with a more generic function format, which expands the single-layer NN controller in Section II, is studied theoretically.

A. NN Controllers with Only Error Term Inputs

If a multi-layer NN controller has only the error term e as input, we use $R(e)$ to represent the NN controller and the control vector u can be expressed as

$$u = R(e) \tag{43}$$

If we substitute (4), (5), and (43) into (1), we can rewrite (1) into the following equation

$$\dot{e} = f(e) = A(e - x_{ref}) - BR(e) \tag{44}$$

Theorem 5. *For a neural dynamic system (44), $e = 0$ is not an equilibrium point except when $x_{ref} = -\frac{1}{k}A^{-1}BR(0)$. Such system will have a non-zero steady-state error, that is $e(\infty) \neq 0$, for any constant reference except one special point $x_{ref} = -\frac{1}{k}A^{-1}BR(0)$.*

Proof: The equilibrium point of (44) is the root of the function $f(e)$. If we substitute $e = 0$ into (6), the function $f(e)$ equals

$$f(e) = -Ax_{ref} - BR(0) \neq 0 \tag{45}$$

The only exception is when $x_{ref} = -\frac{1}{k}A^{-1}BR(0)$. Thus, $e = 0$ is not an equilibrium point of the system (44).

Denote e^* to represent the root, the following equation will be satisfied

$$f(e^*) = A(e^* - x_{ref}) - BR(e^*) = 0 \tag{46}$$

Thus e^* is the equilibrium point of (44) and $e^* \neq 0$, which also means that the system has a non-zero steady-state error $e(\infty) = e^*$. ∎

Theorem 6. *For a neural dynamic system (44), local asymptotic stability and local exponential convergence are guaranteed if the weight matrix and bias vector of the NN satisfy the following condition*

$$Re\left\{\mathrm{eig}\left(A - B\frac{\partial R(e)}{\partial e}|_{e=e^*}\right)\right\} < 0 \tag{47}$$

in which, e^* is the equilibrium point of (44).

Proof: Define $e_n = e - e^*$ and shift the equilibrium point of (44) from e^* to 0. The new system equation will be

$$\dot{e_n} = f(e_n) = A(e_n + e^* - x_{ref}) - BR(e_n + e^*) \tag{48}$$

The right side of (48) is a nonlinear function. Under the definition of Lyapunov stability [18], use the first-order derivative to linearize the system at $e_n = 0$ and obtain the following set of linear equations

$$\dot{e_n} = (\frac{\partial f}{\partial e_n}|_{e_n=0})e_n = Ge_n \tag{49}$$

in which, the closed-loop system matrix G is defined as

$$G = A - B\frac{\partial R(e_n + e^*)}{\partial e_n}|_{e_n=0} = A - B\frac{\partial R(e)}{\partial e}|_{e=e^*} \tag{50}$$

According to Lemma 1, as long as all the eigenvalues of G have the negative real parts, that is $Re\{\text{eig}\,(G)\} < 0$, the system asymptotic stability and exponential convergence are guaranteed. However, as the system (49) is linearized at the equilibrium point, only the local asymptotic stability and local exponential convergence can be guaranteed. ■

Remark 5. In (47), x_{ref} does not exit explicitly. However, e^* is the roots of (46) depending on x_{ref}. So, the system matrix G and eigenvalues λ_i are implicit functions of x_{ref} and thus the reference x_{ref} affects the stability of the system.

B. NN Controllers with Error Integral Inputs

For a multi-layer NN controller containing error term e and error integral s as the inputs, we use $R(e, s)$ to represent the NN controller and the control vector u can be expressed as

$$u = R(e, s) \tag{51}$$

Substituting (4), (5), and (51) into (1), the system equation can be simplified as

$$\dot{e} = A(e - x_{ref}) - BR(e, s) \tag{52}$$

From the definition of error integral s in (21), the following equation can be derived

$$\dot{s} = e \tag{53}$$

Thus combining (52) and (53), a new augmented state-space model can be obtained

$$\begin{cases} \dot{e} &= f_1(e, s) = A(e - x_{ref}) - BR(e, s) \\ \dot{s} &= f_2(e, s) = e \end{cases} \tag{54}$$

Through this conversion, the original n-dimension neural network control system (52) is converted into a $2n$-dimensional system (54).

Theorem 7. *For a neural dynamic system (54), $e = 0$ is an equilibrium point and the system does not have a steady-state error for any constant reference x_{ref}.*

Proof: The equilibrium point of (54) is the roots of the right side function, that is

$$\begin{cases} f_1(e, s) &= A(e - x_{ref}) - BR(e, s) = 0 \\ f_2(e, s) &= e = 0 \end{cases} \tag{55}$$

To satisfy the second equation of (55), e must be 0. Thus the equilibrium point will be (0, s^*), in which s^* satisfies

$$Ax_{ref} + BR(0, s^*) = 0 \tag{56}$$

The equilibrium point of (55) is (0, s^*), which means the system error e converges to 0 while the error integral s converges to s^* when the time goes to infinity. When there is an error integral term s feeding into the input of the NN controllers, it is guaranteed that there is no steady-state error in the system. ■

Theorem 8. *For a neural dynamic system (54), local asymptotic stability and local exponential convergence are guaranteed if the weight matrix and bias vector of the NN satisfy the following condition*

$$Re\left\{\text{eig}\left(\begin{bmatrix} G_{11} & G_{12} \\ I & 0 \end{bmatrix}\right)\right\} < 0 \tag{57}$$

where G_{11} and G_{12} equal

$$G_{11} = A - B\frac{\partial R(e, s)}{\partial e}|_{e=0, s=s^*} \tag{58}$$

$$G_{12} = -B\frac{\partial R(e, s)}{\partial s}|_{e=0, s=s^*} \tag{59}$$

Proof: The equilibrium point of (54) can be shifted from (0, s^*) to (0, 0) using the following conversion:

$$\begin{cases} e &= e \\ s_n &= s - s^* \end{cases} \tag{60}$$

Substituting (60) into (54), the new system equation will be

$$\begin{cases} \dot{e} = f_1(e, s_n) = & A(e - x_{ref}) - BR(e, s_n + s^*) \\ \dot{s_n} = f_2(e, s_n) = & e \end{cases} \tag{61}$$

Under the definition of Lyapunov stability [18] and linearizing (61) at the equilibrium point (0, 0), the system equation will become

$$\begin{bmatrix} \dot{e} \\ \dot{s_n} \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}\begin{bmatrix} e \\ s_n \end{bmatrix} \tag{62}$$

in which, G_{11}, G_{12}, G_{21}, and G_{22} are defined as

$$G_{11} = \frac{\partial f_1(e, s_n)}{\partial e}|_{e=0, s_n=0} = A - B\frac{\partial R(e, s)}{\partial e}|_{e=0, s=s^*} \tag{63}$$

$$G_{12} = \frac{\partial f_1(e, s_n)}{\partial s_n}|_{e=0, s_n=0} = -B\frac{\partial R(e, s)}{\partial s}|_{e=0, s=s^*} \tag{64}$$

$$G_{21} = \frac{\partial f_2(e, s_n)}{\partial e}|_{e=0, s_n=0} = I \tag{65}$$

$$G_{22} = \frac{\partial f_2(e, s_n)}{\partial s_n}|_{e=0, s_n=0} = 0 \tag{66}$$

As G_{11}, G_{12}, G_{21}, and G_{22} are all constants, according to Lemma 1, if the NN weights w and v satisfy the following condition,

$$Re\left\{\text{eig}\left(\begin{bmatrix} G_{11} & G_{12} \\ I & 0 \end{bmatrix}\right)\right\} < 0 \tag{67}$$

the system asymptotic stability and exponential convergence are guaranteed. However, as the system (62) is linearized at the equilibrium point, only the local asymptotic stability and local exponential convergence can be guaranteed. ■

Remark 6. In (62), x_{ref} does not exit explicitly. However, s^* is the equilibrium point of (54) depending on x_{ref}. So, the system matrices G_{11}, G_{12}, G_{21}, and G_{22} and eigenvalues λ_i are implicit functions of x_{ref} and thus the reference x_{ref} affects the stability of the system.

Remark 7. Similar to the conclusion in Remark 4, the weights and the bias vector of the multi-layer NN controller, and the reference together will affect the local stability of the system and thus the local convergence at the equilibrium point. Thus, to guarantee the stable operation of the system, the weights and the bias vectors of the NN controller need to satisfy the stability requirement (47) or (57) for all possible references.

IV. Case Study I : One-Dimensional Neural Network Controllers

In this section, a single-layer NN controller in a one-dimensional state-space model is studied to apply and verify the theories in Section II numerically.

Consider a one-dimensional system of (1) with $A = 2$, $B = 0.5$, and $k = 5$.

A. Single-Layer NN Controllers with Only Error Term Inputs

A single-layer NN controller contains only an error term input and the control action can be expressed as

$$u = k\tanh(we) = 5\tanh(we) \tag{68}$$

Without loss of generality, the bias c is selected as 0.

According to Theorem 1, the system has a steady-state error $e(\infty) = e^*$ for a step reference $x_{ref} = 1$, in which e^* is the root of the following equation

$$\begin{aligned} f(e^*) =& A(e^* - x_{ref}) - kB\tanh(we^* + c) \\ =& 2(e^* - 1) - 5 \times 0.5\tanh(we^* + 0) = 0 \end{aligned} \tag{69}$$

To guarantee local asymptotic stability and local exponential convergence, the NN weight w needs to satisfy the condition specified in Theorem 2. Since we are working with a one-dimensional system, the condition can be simplified further as

$$\begin{aligned} \lambda &= A - kBw\,\mathrm{diag}(1 - \tanh^2(we^* + c)) \\ &= 2 - 5 \times 0.5w[1 - \tanh^2(we^* + 0)] < 0 \end{aligned} \tag{70}$$

Combining (69) and (70), the range of weight w can be obtained. Fig.1 shows the range of w for a step reference $x_{ref} = 1$. When $w = 9.8$, $\lambda = -0.507424234870289 < 0$, which satisfies the stability condition.

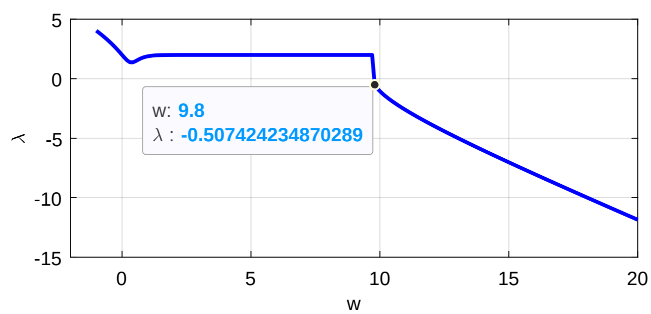

Fig. 1. The eigenvalue λ vs the NN weight w for a step reference $x_{ref} = 1$.

A Simulink model as shown in Fig.2 was built to verify the tracking performance and the steady-state error $e(\infty)$.

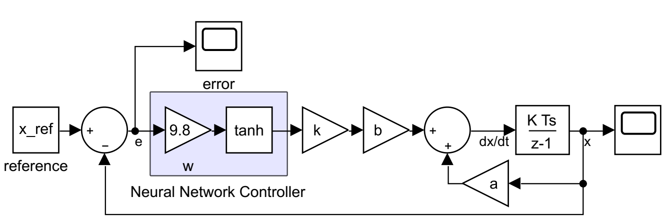

Fig. 2. The Simulink model for the one-dimensional NN controller.

Fig.3 shows the tracking error when NN weight $w = 9.8$ for a step reference. From Fig.3, when $t = 20s$, $e(20s) = -0.184308205223854$, which is pretty close to $e(\infty) = e^* = -0.184308971562349$ from (69) corresponding to $w = 9.8$.

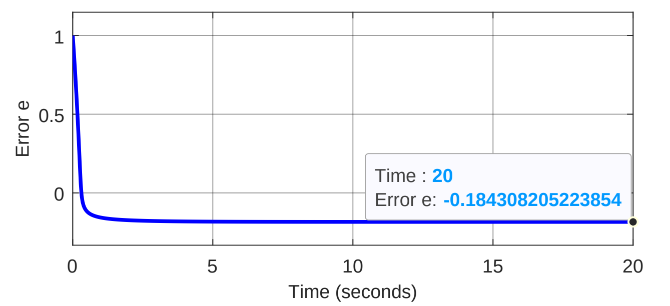

Fig. 3. The tracking error e for a step reference $x_{ref} = 1$ when $w = 9.8$.

B. Adding Error Integral Inputs

To remove the steady-state error, we consider adding the error integral input to the single-layer NN controller as follows

$$u = k\tanh(we + vs) = 5\tanh(9.8e + vs) \tag{71}$$

According to Theorem 3 and (27), the equilibrium point of the system is $[0;\, s^*]$. If v is selected as 1, then s^* equals

$$\begin{aligned} s^* &= -v^{-1}[\operatorname{arctanh}(\frac{1}{k}B^{-1}Ax_{ref}) + c] \\ &= -(1)^{-1}[\operatorname{arctanh}(\frac{1}{5} \times 0.5^{-1} \times 2 \times 1) + 0] \\ &= -1.098612288668110 \end{aligned} \tag{72}$$

The eigenvalues of the NN control system according to Theorem 4 are $\lambda_1 = -6.685377840799436$ and $\lambda_2 = -0.134622159200560$, which satisfy the requirements of local asymptotic stability and local exponential convergence.

A Simulink model as shown in Fig. 4 was built to verify the tracking performance of the NN controller after adding the error integral term.

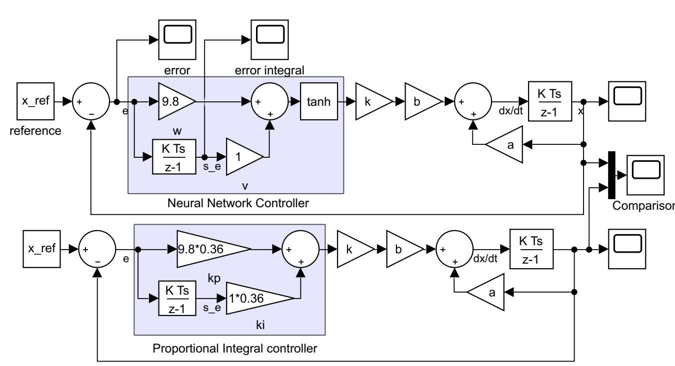

Fig. 4. The Simulink model for a one-dimensional single-layer NN controller and the corresponding PI controller.

The corresponding PI controller with the same eigenvalues was added to the Simulink model to compare steady-state response with the NN controller. To guarantee the designed PI controller to have the same eigenvalues as the single-layer NN controller, we compare (42) and (28)-(30) in Theorem 4, thus set K_p and K_i as

$$K_p = w\,\mathrm{diag}(1 - (\frac{1}{k}B^{-1}Ax_{ref})^2) = 9.8 \times 0.36 \quad (73)$$

$$K_i = v\,\mathrm{diag}(1 - (\frac{1}{k}B^{-1}Ax_{ref})^2) = 1 \times 0.36 \quad (74)$$

Fig. 5 shows the equilibrium point of the one-dimensional single-layer NN control system. When $t = 100s$, $e(100s) = -0.000000214389590$, which is pretty close to theoretical equilibrium point $e(\infty) = 0$. Also, $s(100s) = -1.098610696140083$, which is also very close to theoretical equilibrium point $s(\infty) = s^* = -1.098612288668110$.

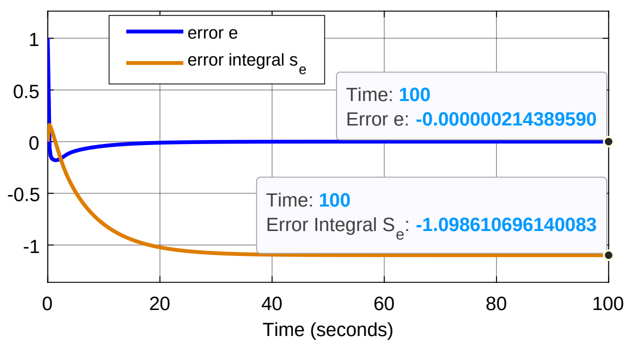

Fig. 5. The equilibrium point of the one-dimensional single-layer NN control system.

To investigate the transient behavior of NN and PI controllers within a neighborhood of their respective equilibrium points, initial values were added to the system state x and the error integral s. As $e = 0$ is the equilibrium point of both NN and PI controllers, x was set as $x(0s) = 0.95$ for both control systems, which means $e(0s) = 1 - 0.95 = 0.05$. According to (41), the equilibrium point s^* for the PI controller is

$$\begin{aligned} s^* &= -K_i^{-1}(\frac{1}{k}B^{-1}Ax_{ref}) \\ &= -(0.36)^{-1}(\frac{1}{5} \times 0.5^{-1} \times 2 \times 1) \\ &= -2.222222222222222 \end{aligned} \quad (75)$$

So the starting points of the error integral s for the NN and PI controllers were set as $s(0s) = -1.098612288668110 + 0.05$ and $s(0s) = -2.222222222222222 + 0.05$, respectively.

Fig. 6 demonstrates the step response for $x_{ref} = 1$ under both NN and PI controllers with starting points from a neighborhood of their respective equilibrium points. Their transient responses are almost the same, which is expected and can be explained by the fact that both NN and PI control systems have exactly the same two eigenvalues.

V. Case Study II: Two-Dimensional Four-Layer Neural Network Controllers in Electric Power Applications

In this section, a four-layer NN controller in a two-dimensional state-space model for renewable energy integration with the electric power grid is studied to apply and verify the theories in Section III numerically.

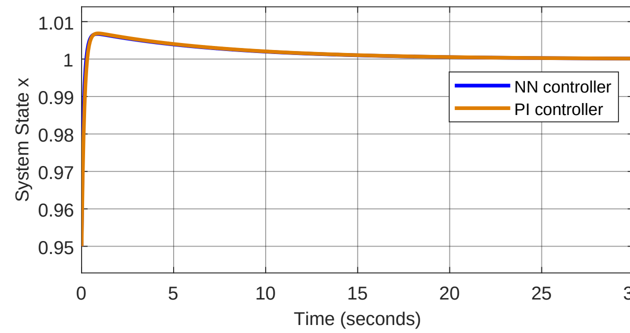

Fig. 6. Step response comparison starting points from a neighborhood of their respective equilibrium points [0; s^*].

A. Grid-Connected Converter

A Grid-Connected Converter (GCC) is a key component that physically connects renewable energy resources such as wind turbines and solar panels to the grid [21], [22], [23], [24], [25]. Fig. 7 shows the schematic of an L filter based GCC, in which a dc-link capacitor is on the left, and a three-phase voltage source, representing the voltage at the Point of Common Coupling (PCC) of the ac system, is on the right.

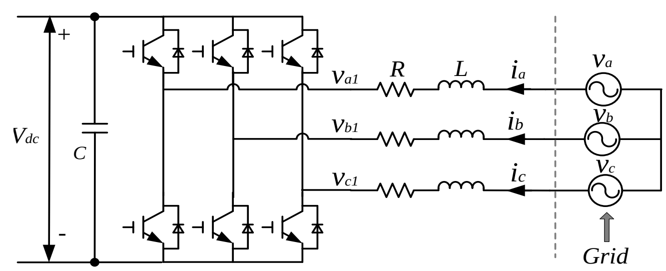

Fig. 7. A grid-connected converter for renewable energy integration.

In the d-q reference frame, the state-space model of the integrated GCC and grid system ([26]) can be expressed as

$$\frac{d}{dt}\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{R}{L} & \omega_s \\ -\omega_s & -\frac{R}{L} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} i_d \\ i_q \end{bmatrix}}_{i_{dq}} + \underbrace{\begin{bmatrix} -\frac{1}{L} & 0 \\ 0 & -\frac{1}{L} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} V_{d1} - V_d \\ V_{q1} - Vq \end{bmatrix}}_{u_{dq}} \quad (76)$$

where ω_s is the angular frequency of the grid voltage, and L and R represent the inductance and resistance of the grid filter respectively, the system states are $i_{dq} = [i_d; i_q]$, the grid PCC voltages $V_{dq} = [V_d; V_q]$ are normally constants, and $V_{dq1} = [V_{d1}; V_{q1}]$ are the converter output voltages that are specified by the current controller outputs, and the control vector $u_{dq} = V_{dq1} - V_{dq}$.

Table I specifies all system parameters in a lab experiment setup [27]. Using the parameters from Table I, $V_{dq} = [V_{dc}; 0] = [20; 0]$ and $k_{pwm} = \sqrt{3/2}\frac{V_{dc}}{2} = 30.618621784789724$.

B. Four-Layer NN Controller

The NN current controller $N(e_{sdq}, s_{sdq}, w)$ is a function of the error e_{dq}, the error integral s_{dq} and the weights w. As the ratio of the converter output voltage V_{dq1} to the outputs of the

TABLE I
THE L FILTER BASED GCC SYSTEM PARAMETERS

Symbol	Quantity	Value	Unit
V_g	test grid voltage (rms)	20	V
f	nominal grid frequency	60	Hz
V_{dc}	DC-link voltage	50	V
L	grid side inductor	25	mH
R	grid side resistor	0.25	Ω

current controller is the gain of the Pulse-Width-Modulation (PWM) k_{pwm}[28], the control action u_{dq} is then expressed by

$$u_{dq} = R(e_{dq}, s_{dq}) = V_{dq1} - V_{dq} = k_{pwm} N(e_{dq}, s_{dq}, w) - V_{dq} \quad (77)$$

The structure of the four-layer NN controller ([10], [11]) is shown in Fig. 8.

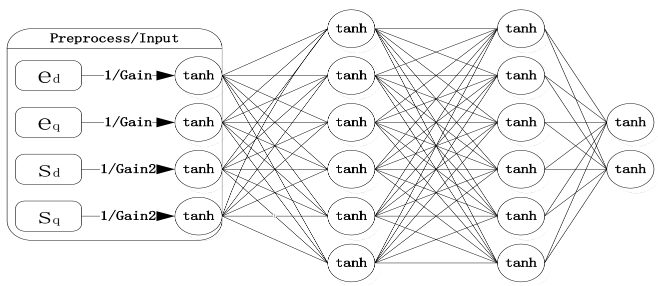

Fig. 8. The structure of the four-layer NN controller used in the GCC for grid integration.

The function format of the four-layer NN controller $R(e_{dq}, s_{dq}, w)$ can be represented as

$$N(e_{dq}, s_{dq}, w) = \tanh\left(w_3\begin{bmatrix}\tanh\left(w_2\begin{bmatrix}\tanh\left(w_1\begin{bmatrix}\tanh\begin{bmatrix}\frac{e_{dq}}{Gain}\\ \frac{s_{dq}}{Gain2}\end{bmatrix}\\ -1\end{bmatrix}\right)\\ -1\end{bmatrix}\right)\\ -1\end{bmatrix}\right) \quad (78)$$

where w_1,w_2, and w_3 represent the weights from the input layer to the first hidden layer, the first to the second hidden layer and the second hidden layer to the output layer respectively. The biases of each layer have been incorporated into w_1, w_2, and w_3 to simplify the weight updating process.

The four-layer NN controller was trained by the LMBP [29], [30], [31] and the FATT algorithm [32]. For the four-layer NN controller, its weight parameters $Gain_1$, $Gain_2$. w_1, w_2, and w_3 are listed in Table II.

The equilibrium point of the system is (0, s^*_{dq}). According to Theorem 7 and (56), s^*_{dq} satisfies the following function

$$\underbrace{\begin{bmatrix}-\frac{R}{L} & \omega_s\\ -\omega_s & -\frac{R}{L}\end{bmatrix}}_{A} i_{dq\_ref} + \underbrace{\begin{bmatrix}-\frac{1}{L} & 0\\ 0 & -\frac{1}{L}\end{bmatrix}}_{B}[k_{pwm}N\big(0, s^*_{dq}, w\big) - V_{dq}] = 0 \quad (79)$$

According to (58) and (59) in Theorem 8, G_{11} and G_{12} can be calculated as

$$\begin{aligned} G_{11} &= A - B\frac{\partial R(e_{dq}, s_{dq})}{\partial e_{dq}}|_{e_{dq}=0, s_{dq}=s^*_{dq}} \\ &= A - k_{PWM}B\frac{\partial N(e_{dq}, s_{dq}, w)}{\partial e_{dq}}|_{e_{dq}=0, s_{dq}=s^*_{dq}} \quad (80)\end{aligned}$$

$$\begin{aligned} G_{12} &= -B\frac{\partial R(e_{dq}, s_{dq})}{\partial s_{dq}}|_{e_{dq}=0, s_{dq}=s^*_{dq}} \\ &= -k_{PWM}B\frac{\partial N(e_{dq}, s_{dq}, w)}{\partial s_{dq}}|_{e_{dq}=0, s_{dq}=s^*_{dq}} \quad (81)\end{aligned}$$

The details of calculating G_{11} and G_{12} are listed in Appendix A.

Given the current reference $i_{dq\_ref} = [1; 0]$, the corresponding four eigenvalues can be calculated and are listed in Table III.

C. Corresponding PI controller

A PI controller was designed to have exactly the same four eigenvalues as those of the four-layer NN controller for comparison. Table III lists the target four eigenvalues for the single-layer NN controller and the PI controller.

To guarantee the designed PI controller to have the same eigenvalues as the four-layer NN controller, we compare (42) and (80) and (81), thus set K_p and K_i as

$$K_p = \frac{\partial N(e_{dq}, s_{dq}, w)}{\partial e_{dq}}|_{e_{dq}=0, s_{dq}=s^*_{dq}} \quad (82)$$

$$K_i = \frac{\partial N(e_{dq}, s_{dq}, w)}{\partial s_{dq}}|_{e_{dq}=0, s_{dq}=s^*_{dq}} \quad (83)$$

Table IV lists the values of K_p and K_i. Unlike the conventional one-dimensional PI controller with a scalar proportional gain and a scalar integral gain, the generalized PI controller shown in Table IV has cross-coupling terms and are in a more generalized gain matrix format, which has better and stronger performance than the conventional one-dimensional PI controller.

TABLE IV
THE PARAMETERS OF THE CORRESPONDING PI CONTROLLER

K_p	$\begin{bmatrix}-0.344022164281883 & 0.727142679990575\\ -0.754209007295918 & -1.18991558063817\end{bmatrix}$
K_i	$\begin{bmatrix}-2.3221654488264 & 196.559123343741\\ -153.905082611539 & -54.8126346765554\end{bmatrix}$

D. Corresponding Single-layer NN controller

For the single-layer NN controller design, the bias vector c was selected as zeros to simplify the design process. We compare (42) and (29) and (30), and thus set weights w and v as

$$w = [\mathrm{diag}(1 - (\frac{1}{k_{pwm}}B^{-1}Ax_{ref})^2)]^{-1}K_p \quad (84)$$

$$v = [\mathrm{diag}(1 - (\frac{1}{k_{pwm}}B^{-1}Ax_{ref})^2)]^{-1}K_i \quad (85)$$

Table V lists the values of w and v for the single-layer NN controller.

TABLE II
THE WEIGHT PARAMETERS OF THE FOUR-LAYER NN CONTROLLER

Gain1	0.5
Gain2	0.5
W1	$\begin{bmatrix} 0.105118602490750 & -0.869195807768507 & 3.910726574451215 & 3.829650558215191 & 0.043137396666237 \\ 0.805253127771219 & 0.116082719739100 & 5.081202415079452 & -1.666910901747036 & 1.246185328567662 \\ 0.142117134906564 & 0.272375503071100 & 4.040974596086221 & 2.048223953909149 & 0.145066179115310 \\ -0.395272323007882 & -1.422986921530577 & 4.255131423501219 & 5.561432608781822 & 0.000251202007245 \\ -0.277224746255928 & 0.699935881635053 & 1.636989905274748 & 2.678881970530615 & -0.109536124233592 \\ 0.376127545234790 & 1.285716734931245 & -2.973060687194107 & 8.095548964772654 & 0.028991336931277 \end{bmatrix}$
W2	$\begin{bmatrix} 1.440539114493213 & -0.272530718390058 & 0.527886890221929 & 1.371222680616433 & 2.255139286184510 & 1.394844625523901 & 0.344937425499452 \\ 3.378981724654836 & 0.198608148623109 & 3.459270721071458 & -1.911027029429807 & 0.224751908404989 & -1.002210347176314 & -0.475383054734012 \\ -0.359022430219653 & 1.217655164464906 & 3.145578151429633 & 1.863120732645271 & 3.708974043074285 & -0.096082441939513 & 1.566135015097376 \\ 4.031099888420788 & -2.685187585928909 & -2.749868864734965 & 2.748659888667571 & 2.439552173754654 & 5.660170953027147 & 0.925728746264457 \\ -0.622152942608992 & 0.732064874764135 & 4.212370496471141 & -4.081216558783956 & 1.547382976445135 & -6.456534076312040 & -0.817547511050558 \\ 1.522500050952956 & -1.036035004009775 & 1.703072013081991 & 0.534432723278869 & 0.630762934216796 & 1.093038633050528 & 0.074249030990423 \end{bmatrix}$
W3	$\begin{bmatrix} -1.711155394435648 & -0.447196877031189 & -2.614508912856286 & -5.956955188009836 & 0.958589844957509 & 1.641209174573893 & -2.440547421725287 \\ 1.317977684822903 & 1.038607717509133 & 2.191677954355899 & -0.515283801531746 & 0.973061014722440 & 3.038069686362197 & -1.386654203297988 \end{bmatrix}$

TABLE III
EIGENVALUES

Control method	λ			
Four-layer NN	-802.233078413318 +1100.64099842807i	-802.233078413318 - 1100.64099842807i	-147.10811464909 + 54.0774179743671i	-147.10811464909 - 54.0774179743671i
Single-layer NN	-802.233078413318 +1100.64099842807i	-802.233078413318 - 1100.64099842807i	-147.10811464909 + 54.0774179743671i	-147.10811464909 - 54.0774179743671i
PI	-802.233078413318 +1100.64099842807i	-802.233078413318 - 1100.64099842807i	-147.10811464909 + 54.0774179743671i	-147.10811464909 - 54.0774179743671i

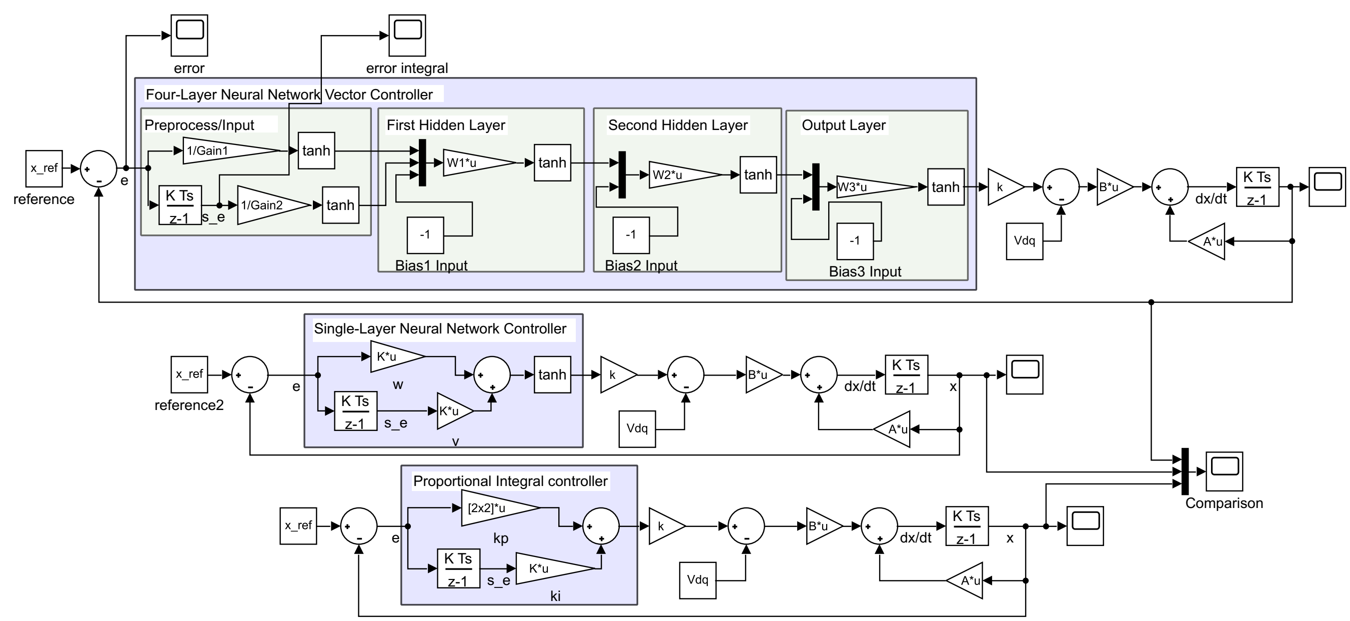

Fig. 9. The Simulink model for three controllers with the same four eigenvalues: four-layer NN controller, single-layer NN controller, and PI controller.

TABLE V
THE PARAMETERS OF THE CORRESPONDING SINGLE-LAYER NN CONTROLLER

w	$\begin{bmatrix} -0.589146302572011 & 0.803249086937196 \\ -1.291601222678247 & -1.314458124906575 \end{bmatrix}$
v	$\begin{bmatrix} -3.97676466861468 & 217.131988947592 \\ -263.566187826589 & -60.5496004678695 \end{bmatrix}$

TABLE VI
THE EQUILIBRIUM POINTS

Control method	$e_{dq}(\infty)$		$s_{dq}(\infty)$	
Four-layer NN	0	0	0.000570367398365	0.000995539550846
Single-layer NN	0	0	0.000827793898875	0.003291399362787
PI	0	0	0.000394111939873	0.003538453922689

E. Step Response Comparison

A Simulink model as shown in Fig. 9 was built to compare all three controllers.

Fig.10 shows the tracking error e_{dq} for a step response $i_{dq} = [1; 0]$. At time $t = 0.1s$, $e_{dq} = [-1.886e - 7; 6.882e - 08]$, which is already very close to the equilibrium point [0;0].

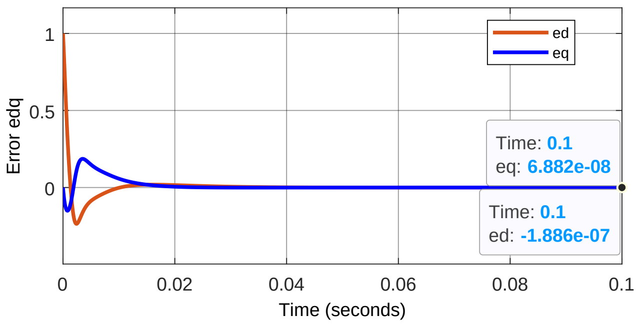

Fig. 10. The tracking error e_{dq} for step response $i_{dq} = [1; 0]$.

Fig.11 shows the corresponding steady-state of s_{dq} for a step response of $i_{dq} = [1; 0]$. At time $t = 0.1s$, $s_{dq} = [0.000570368236706; 0.000995538938232]$, which has 8 significant bits the same as the equilibrium point [0.000570367398365;0.000995539550846] in Table VI.

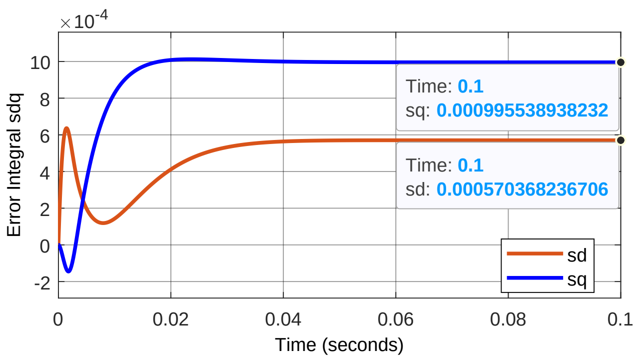

Fig. 11. The error integral s_{dq} for step response $i_{dq} = [1; 0]$.

Table VI lists the equilibrium points for all three control methods. As all three control methods have the error integral inputs s_{dq}, the equilibrium points for the e_{dq} are all zeros, which is the steady-state error of $e_{dq}(\infty) = 0$. For the error integral s_{dq}, they all converge to their respective equilibrium points as each control method has different weights or parameters.

To evaluate and compare the steady-state behaviors of all three control methods close to their equilibrium points, starting points were given to s_{dq} while $i_{dq} = [0; 0]$ and $e_{dq} = [1; 0]$ were kept unchanged in the simulation. The staring points for s_{dq} were set as $s_{dq}(0s) = s_{dq}(\infty) - [0.001; 0]$, which was to study the behavior within a small neighborhood of their equilibrium points $s_{dq}(\infty)$. starting points of s_{dq} are listed in Table VII.

TABLE VII
NEW STARTING POINTS OF s_{dq}

Control method	$s_{dq}(0s) = s_{dq}(\infty) - [0.001; 0]$	
Four-layer NN	0.000570367398365 - 0.001	0.000995539550846
Single-layer NN	0.000827793898875 - 0.001	0.003291399362787
PI	0.000394111939873 - 0.001	0.003538453922689

Fig. 12 shows the step response under this condition, which have pretty similar responses within a small neighborhood of their equilibrium points and verify the fact that they all have the same four eigenvalues.

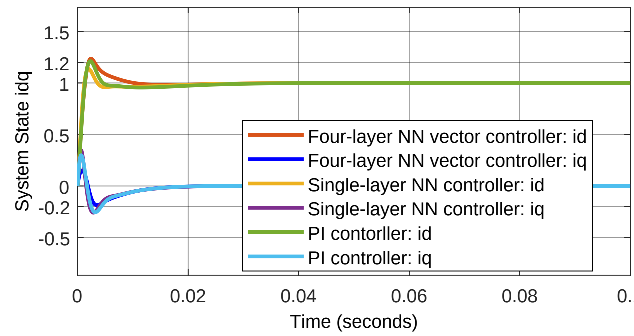

Fig. 12. Step response comparison with starting points from a neighborhood of their respective equilibrium points s^*_{dq}.

Thus, it is expected that when the size of the neighborhood around the equilibrium points is small enough, all three control methods will demonstrate identical transient behaviors due to the fact they all have the same four eigenvalues.

VI. Conclusion

This paper proves that if the neural network controllers only have error terms as inputs, the control system has a non-zero steady-state error for any constant reference in general except for one special reference point. Adding an error integral term to the inputs of the NN controller is sufficient to eliminate the steady-state error for any constant reference.

This paper also provides a simple way of using eigenvalues of the NN control system to evaluate local stability and local convergence for reference tracking, which has almost the same transient behavior as the corresponding generalized PI controllers with the same eigenvalues and helps practical engineers understand the essence of NN controllers.

Unlike the generalized PI controller, the reference affects the stability and convergence of the NN control system. Therefore, for a NN controller, the training of the NN is needed to assure that for all the representative references in the targeted and acceptable region, the well-trained NN weights would satisfy the stable network eigenvalue requirement, which is a common requirement for the development of an NN controller.

Appendix A
Derivation of G_{11} and G_{12} for the Four-Layer NN Controller

To simplify the derivation process, define o_e, o_s, o_1, o_2, and o_3 as follows:

$$o_e = \tanh(e_{dq}/Gain1)|_{e_{dq}=[0;0]} = \tanh([0;0]/Gain1) \quad (86)$$

$$o_s = \tanh(s_{dq}/Gain2)|_{s_{dq}=s^*_{dq}} = \tanh(s^*_{dq}/Gain2) \quad (87)$$

$$o_1 = \tanh(w_1[o_e; o_s; -1]) \quad (88)$$

$$o_2 = \tanh(w_2[o_1; -1]) \quad (89)$$

$$o_3 = N(e_{dq}, s_{dq}, w) = \tanh(w_3[o_2; -1]) \quad (90)$$

Then

$$\frac{\partial N(e_{dq},s_{dq},w)}{\partial e_{dq}}\Big|_{e_{dq}=0,s_{dq}=s^*_{dq}} = \frac{\partial o_3}{\partial o_2}\frac{\partial o_2}{\partial o_1}\frac{\partial o_1}{\partial o_e}\frac{\partial o_e}{\partial e_{dq}}\Big|_{e_{dq}=0,s_{dq}=s^*_{dq}}$$
$$= [\mathrm{diag}(1-o_3^2)w_3(:,1:6)][\mathrm{diag}(1-o_2^2)w_2(:,1:6)]$$
$$*[\mathrm{diag}(1-o_1^2)w_1(:,1:2)][\mathrm{diag}((1-o_e^2)/Gain1)] \quad (91)$$
$$\frac{\partial N(e_{dq},s_{dq},w)}{\partial s_{dq}}\Big|_{e_{dq}=0,s_{dq}=s^*_{dq}} = \frac{\partial o_3}{\partial o_2}\frac{\partial o_2}{\partial o_1}\frac{\partial o_1}{\partial o_s}\frac{\partial o_s}{\partial s_{dq}}\Big|_{e_{dq}=0,s_{dq}=s^*_{dq}}$$
$$= [\mathrm{diag}(1-o_3^2)w_3(:,1:6)][\mathrm{diag}(1-o_2^2)w_2(:,1:6)]$$
$$*[\mathrm{diag}(1-o_1^2)w_1(:,3:4)][\mathrm{diag}((1-o_s^2)/Gain2)] \quad (92)$$

Substitute (91) and (92) into (80) and (81). Thus G_{11} and G_{12} can be obtained.

References

[1] R. E. Bellman, *Dynamic Programming*. Princeton, New Jersey, USA: Princeton University Press, 1957.

[2] S. N. Balakrishnan and V. Biega, "Adaptive-critic-based neural networks for aircraft optimal control," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 4, pp. 893–898, Jul./Aug. 1996.

[3] S. Al-Dabooni and I. Donald C. Wunsch, "Online model-free n-step hdp with stability analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1255–1269, Apr. 2020.

[4] ——, "An improved n-step value gradient learning adaptive dynamic programming algorithm for online learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1155–1169, Apr. 2020.

[5] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Transactions on Neural Networks*, vol. 8, no. 5, pp. 997–1007, Sep. 1997.

[6] F. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 39–47, May 2009.

[7] G. K. Venayagamoorthy, R. G. Harley, and D. C. Wunsch, "Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 764–773, May 2002.

[8] S. Li, M. Fairbank, D. C. Wunsch, and E. Alonso, "Vector control of a grid-connected rectifier/inverter using an artificial neural network," in *Proc. IEEE World Congress on Computational Intelligence*, Brisbane, Australia, Jun. 2012, pp. 1–7.

[9] S. Li, M. Fairbank, C. Johnson, D. C. Wunsch, E. Alonso, and J. L. Proaño, "Artificial neural networks for control of a grid-connected rectifier/inverter under disturbance, dynamic and power converter switching conditions," *IEEE Transactions on Neural Networks And Learning Systems*, vol. 25, no. 4, pp. 738–750, Apr. 2014.

[10] X. Fu, S. Li, and I. Jaithwa, "Implement optimal vector control for LCL-filter-based grid-connected converters by using recurrent neural networks," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 7, pp. 4443 – 4454, Jul. 2015.

[11] X. Fu and S. Li, "Control of single-phase grid-connected converters with LCL filters using recurrent neural network and conventional control methods," *IEEE Transactions on Power Electronics*, vol. 31, no. 7, pp. 5354 – 5364, Jul. 2016.

[12] S. Li, H. Won, X. Fu, M. Fairbank, D. C. Wunsch, and E. Alonso, "Neural-network vector controller for permanent-magnet synchronous motor drives: Simulated and hardware-validated results," *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 3218–3230, Jul. 2020.

[13] J. M. Bentez, J. L. Castro, and I. Requena, "Are artificial neural networks black boxes?" *IEEE Transactions on Neural Networks*, vol. 8, no. 5, pp. 1156–1164, Sept. 1997.

[14] N. E. Barabanov and D. V. Prokhorov, "Stability analysis of discrete-time recurrent neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 13, no. 2, pp. 292–303, Mar. 2002.

[15] H. Zhang, Z. Wang, and D. Liu, "Global asymptotic stability of recurrent neural networks with multiple time-varying delays," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 19, no. 5, pp. 855–873, May 2008.

[16] G.-H. Liu and E. Theodorou, "Deep learning theory review: An optimal control and dynamical systems perspective," *ArXiv*, vol. abs/1908.10920, 2019.

[17] H. K. Khalil, *Nonlinear Control*. Upper Saddle River, New Jersey: Pearson Education, 2015.

[18] H. Nijmeijer and A. van der Schaft, *Nonlinear Dynamical Control Systems*. New York City, New York: Springer, 1990.

[19] N. S. Nise, *Control Systems Engineering, 7th Edition*. Hoboken, New Jersey: John Wiley & Sons, 2014.

[20] R. C. Dorf and R. H. Bishop, *Modern Control Systems, 13th Edition*. Upper Saddle River, New Jersey: Pearson Education, 2016.

[21] G. Buticchi, D. Barater, E. Lorenzani, and G. Franceschini, "Digital control of actual grid-connected converters for ground leakage current reduction in pv transformerless systems," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 3, pp. 563–572, August 2012.

[22] J. Shen, H. Jou, and J. Wu, "Novel transformerless grid-connected power converter with negative grounding for photovoltaic generation system," *IEEE Transactions on Power Electronics*, vol. 27, no. 4, pp. 1818–1829, April 2012.

[23] X. Zhou, S. Lukic, S. Bhattacharya, and A. Huang, "Design and control of grid-connected converter in bi-directional battery charger for plug-in hybrid electric vehicle application," in *Proc. IEEE Vehicle Power and Propulsion Conference (VPEC'09)*, Dearborn, Michigan, USA, September 2009, pp. 1716–1721.

[24] M. Hanif, V. Khadkikar, W. Xiao, and J. L. Kirtley, "Two degrees of freedom active damping technique for LCL filter-based grid connected pv systems," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 6, pp. 2795–2803, Jun. 2014.

[25] M. Liserre, F. Blaabjerg, and S. Hansen, "Design and control of an LCL filter-based three-phase active rectifier," *IEEE Transactions on Industrial Applications*, vol. 41, no. 5, pp. 1281–1291, Oct. 2005.

[26] S. Li, M. Fairbank, C. Johnson, D. C. Wunsch, E. Alonso, and J. L. Proaño, "Artificial neural networks for control of a grid-connected rectifier/inverter under disturbance, dynamic and power converter switching conditions," *IEEE Transactions on Neural Networks And Learning Systems*, vol. 25, no. 4, pp. 738–750, Apr. 2014.

[27] X. Fu and S. Li, "A novel neural network vector control for single-phase grid-connected converters with L, LC and LCL filters," *Energies*, vol. 9, no. 5, p. 328, May 2016.

[28] N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics: Converters, Applications, and Design*, 3rd ed. Chichester, West Sussex, United Kingdom: John Wiley and Sons, 2002.

[29] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, November 1994.

[30] D. W. Marquardt, "An algorithm for least-squares estimation of non-linear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, Jun. 1963.

[31] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design*. Boston, USA: PWS Publishing, 2002.

[32] X. Fu, S. Li, M. Fairbank, D. C. Wunsch, and E. Alonso, "Training recurrent neural networks with the Levenberg-Marquardt algorithm for optimal control of a grid-connected converter," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 9, pp. 1900 – 1912, Sept. 2015.